CLAIMS

1. (Currently Amended) A method to prevent starvation in a switched system comprising a distributed bus arbiter that varies a time between a request for the switched system resources as a function of a number of requests the system indicates to be retried, wherein the step of varying the time in the switched system is a function of a plurality of requests, but does not keep track of latency times of any selected processor or proxy processor.

2. (Original) The method of Claim 1 further comprising the bus arbiter first increasing the time between requests until the system no longer indicates to retry the request and the bus arbiter then decreasing the time between requests after a number of requests processed without the system indicating to retry any requests.

3. (Currently Amended) A method for preventing starvation in a switched system with a distributed bus arbiter, the method comprising:

a) the bus arbiter increasing a time between a request from a microprocessor connected to the switched system to use resources of a switch until the switch can process the request; and

b) the bus arbiter decreasing the time between the requests from the microprocessors connected to the switched system to use resources of the switch as a function of some number of requests processed without the bus arbiter having to increase the time between requests from the microprocessors connected to the switched system, wherein the steps of increasing or decreasing the time in the switched system is a

function of a plurality of requests, but does not keep track of latency times of any selected processor or proxy processor.

4. (Currently Amended)) A method for avoiding livelocks in a switched system with a distributed bus arbiter, whereby bus request logic varies a time between requests for resources of the switched system as a function of switch retries, the method comprising:

a) defining a first parameter that specifies a number of bus clocks to wait between requests from a microprocessor connected to the switched system;

b) defining a second parameter that specifies a number of sequential responses to be received from a system switch indicating it has a sufficient number of resources to process the request issued from the microprocessor connected to the switched system;

c) issuing at least one request from the microprocessor to the system switch;

d) responsive to a signal from the system switch indicating it lacks resources to process a command, increasing the number of bus clocks to wait between issuance of requests from microprocessors by the number of bus clocks defined as the first parameter;

e) waiting the increased number of bus clocks between issuance of requests to the switch;

f) responsive to a signal from the system switch indicating it has resources to process a command, increasing a counter counting the number of signals from the system switch indicating it has resources to process a command;

g) responsive to a signal from the system switch indicating it lacks resources to process a command, resetting the value in the counter to zero;

h) comparing the value in the counter to the second parameter; and when the value in the counter is equal to the value of the second parameter, decreasing the number of bus clocks between issuance of requests to the switch from microprocessors connected to the switched system by the amount equal to the number of bus clocks defined as the first parameter and resetting the value in the counter to zero, wherein the steps of increasing or decreasing the time in the switched system is a function of a plurality of requests, does not keep track of latency times of any selected processor or proxy processor.

5. (Original) The method of Claim 4 wherein the first parameter is defined at system power-on.

6. (Original) The method of Claim 4 wherein the second parameter is defined at system power-on.

7. (Currently Amended) A distributed bus arbiter in a switched system that prevents livelocks, the bus arbiter varying the time between a request for the switched system resources as a function of a number of requests the switched system indicates to be retried, retried, wherein varying the time in the switched system is a function of a plurality of requests, but does not keep track of latency times of any selected processor or proxy processor.

8. (Currently Amended) The apparatus of Claim 7 further comprising the bus arbiter first increasing the time between requests until the system no longer indicates to retry the request and the bus arbiter then decreasing the time between requests after a number of requests have been processed without the system indicating to retry any requests.

9. (Currently Amended) A distributed bus arbiter for use in a switched system that prevents livelocks, the system comprising:

a) the bus arbiter increasing a time between a request from a microprocessor connected to the switched system to use resources of a switch until the switch can process the request; and

b) the bus arbiter decreasing the time between the requests from the microprocessors connected to the switched system to use resources of the switch as a function of some number of requests processed without the bus arbiter having to increase the time between requests from the microprocessors connected to the switched system, wherein increasing or decreasing the time in the switched system is a function of a plurality of requests, but does not keep track of latency times of any selected processor or proxy processor.

10. (Currently Amended) A distributed bus arbiter apparatus for preventing livelocks in a switched system by varying a time between requests for system resources as a function of switch retries, the apparatus comprising:

a) means for defining a first parameter to specify a number of bus clocks to wait between processing requests from a microprocessor connected to the switched system;

b) means for defining a second parameter to specify a number of sequential responses to be received from a system switch indicating it has a sufficient number of resources to process the request issued from the microprocessor connected to the switched system;

c) means for issuing at least one request from the microprocessor to the system switch;

d) means for issuing a signal from the system switch indicating it lacks resources to process a command,

e) responsive to the signal from the system indicating it lacks resources to process a command, means for increasing the number of bus clocks to wait between issuance of requests by an amount equal to the first parameter;

f) means for waiting the increased number of bus clocks between issuance of requests to the switch;

g) means, responsive to a signal from the system switch indicating it has resources to process a command, for increasing a counter counting the number of signals from the system switch indicating it has resources to process a command;

h) means, responsive to a signal from the system switch indicating it lacks resources to process a command, for resetting the value in the counter to zero;

i) means for comparing the value in the counter to the second parameter; and

j) means for decreasing the number of bus clocks between issuance of requests to the switch from microprocessors by the amount equal to the number of bus clocks defined as the first parameter and resetting the value in the counter to zero when the value in the counter is equal to the value of the second parameter, wherein the means for increasing or decreasing the time in the switched system is a function of a plurality of requests, but does not keep track of latency times of any selected processor or proxy processor.

11. (Original) The apparatus of Claim 10 further comprising means for defining the first parameter at system power-on.

12. (Original) The apparatus of Claim 10 further comprising means for defining the second parameter at system power-on.

13. (Currently Amended) A computer program product for preventing livelocks in a switched system with a distributed bus arbiter, the computer program comprising:

a) computer program code for increasing a time between a request from a microprocessor connected to the switched system to use resources of a switch until the switch can process the request; and

b) computer program code for decreasing the time between the requests from the microprocessors to use resources of the switch as a function of some number of requests processed without the bus arbiter having to increase the time between requests from the microprocessors connected to the switched system, wherein the computer

program code for increasing or decreasing the time in the switched system is a function of a plurality of requests, but does not keep track of latency times of any selected processor or proxy processor.

14. (Currently Amended) A computer program product for preventing livelocks in a switched system by varying a time between requests for system resources as a function of the number of requests to be retried due to lack of system resources, the computer program product comprising:

a) computer program code for defining a first parameter to specify a number of bus clocks to wait between processing requests from a microprocessor connected to the switched system;

b) computer program code for defining a second parameter to specify a number of sequential responses to be received from a system switch indicating it has a sufficient number of resources to process the request issued from the microprocessor;

c) computer program code for issuing requests from the microprocessor to the system switch;

d) computer program code for the system switch to indicate it lacks resources to process a command,

e) computer program code for increasing the number of bus clocks to wait between issuance of requests by an amount equal to the first parameter responsive to the system switch indicating it lacks resources to process a command;

f) computer program code for the system to increase a counter counting the number of signals from the system switch responsive to the system indicating it has resources to process a command;

g) computer program code for resetting the value in the counter to zero responsive to a signal from the system switch that it lacks resources to process a command;

h) computer program code for comparing the value in the counter to the second parameter;

i) computer program code for decreasing the number of bus clocks between issuance of requests to the switch from microprocessors by the amount equal to the number of bus clocks defined as the first parameter; and

j) computer program code for resetting the value in the counter to zero when the value in the counter is equal to the value of the second parameter, wherein the computer program code for increasing or decreasing the time in the switched system employs a plurality of requests, but does not keep track of latency times of any selected processor or proxy processor.